

Refine Search

Search Results -

Terms	Documents
L9 AND html	0

Database: [US Pre-Grant Publication Full-Text Database](#)
[US Patents Full-Text Database](#)
[US OCR Full-Text Database](#)
[EPO Abstracts Database](#)
[JPO Abstracts Database](#)
[Derwent World Patents Index](#)
[IBM Technical Disclosure Bulletins](#)

Search:

Search History

DATE: Sunday, November 06, 2005 [Printable Copy](#) [Create Case](#)

Set Name **Query**
side by side

DB=USPT; PLUR=NO; OP=OR

		Hit Count	Set Name
			result set
<u>L10</u>	L9 AND html	0	<u>L10</u>
<u>L9</u>	L8 AND trigger and token and pars\$3 and lex\$	7	<u>L9</u>
<u>L8</u>	L7 OR l6 OR l4 OR l3 OR l2 OR l1	448	<u>L8</u>
<u>L7</u>	715/542.ccls.	110	<u>L7</u>
<u>L6</u>	707/534	99	<u>L6</u>
<u>L5</u>	707/534.ccls	0	<u>L5</u>
<u>L4</u>	717/113.ccls.	104	<u>L4</u>
<u>L3</u>	717/111.ccls.	53	<u>L3</u>
<u>L2</u>	717/110.ccls.	94	<u>L2</u>
<u>L1</u>	717/112.ccls.	20	<u>L1</u>

END OF SEARCH HISTORY

Hit List

First Hit	Clear	Generate Collection	Print	Fwd Refs	Bkwd Refs
Generate OACs					

Search Results - Record(s) 1 through 7 of 7 returned.

1. Document ID: US 6314559 B1

L9: Entry 1 of 7

File: USPT

Nov 6, 2001

US-PAT-NO: 6314559

DOCUMENT-IDENTIFIER: US 6314559 B1

** See image for Certificate of Correction **

TITLE: Development system with methods for assisting a user with inputting source code

DATE-ISSUED: November 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sollich; Peter Franz Valentin	Santa Cruz	CA		

US-CL-CURRENT: 717/111

ABSTRACT:

A visual development system having an interface which assists a user with input of source code expressions and statements during creation of a computer program is described. The interface includes an Integrated Development Environment (IDE) interface having a code editor with "Code Completion" and "Code Parameter" features for displaying context sensitive pop-up windows within a source code file. Code Completion is implemented at the user interface level by displaying a Code Completion dialog box after the user enters a record or class name followed by a period. For a class, the dialog lists the properties, methods and events appropriate to the class. For a record or structure, the dialog lists the data members of the record. To complete entry of the expression, the user need only select an item from the dialog list, whereupon the system automatically enters the selected item in the code. Code completion also operates during input of assignment statements. When the user enters an assignment statement for a variable and presses a hot key (e.g., <ctrl><space_bar>), a list of arguments valid for the variable is displayed. Here, the user can simply select an argument to be entered in the code. Similarly, the user can bring up a list of arguments when typing a procedure, function, or method call and needs to add an argument. In this manner, the user can view the required arguments for a method as he or she enters a method, function, or procedure call.

5 Claims, 35 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 34

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	RDNC	Dra
------	-------	----------	-------	--------	----------------	------	-----------	--------	------	-----

2. Document ID: US 5857212 A

L9: Entry 2 of 7

File: USPT

Jan 5, 1999

US-PAT-NO: 5857212

DOCUMENT-IDENTIFIER: US 5857212 A

TITLE: System and method for horizontal alignment of tokens in a structural representation program editor

DATE-ISSUED: January 5, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van De Vanter; Michael L.	Mountain View	CA		

US-CL-CURRENT: 715/519; 715/527, 715/530, 715/531, 717/111

ABSTRACT:

An editor for structurally represented computer programs transforms user-entered text on-the-fly into a stream of tokens that constitute words of the program under edit. Each token is classified as one of group of extended lexemes, and based upon token stream information the editor prettyprint displays the program as the user types. Prettyprinting involves typesetting each token in a visually distinct manner and displaying a varying amount of visual inter-token whitespace between the tokens, based upon token lexical type. The program may be user-edited from the prettyprinted display as though the program were internally represented as text. Cursor position and display appearance depend on the lexical types of tokens adjacent the cursor. To improve aesthetics of the prettyprinted display, a user may insert one or more alignment markers into lines of associated text. The presence of such marker(s) forces horizontal alignment between associated text lines containing such markers. The presence, number, and occurrence of such markers in associated lines of text is noted, and the pixel distance from a boundary edge to the first occurring marker in each line is calculated. The maximum such distance determines relative position of the first alignment marker. Pixel units are added to the whitespace gap preceding the first marker in the other associated lines to force such markers into alignment with the marker whose position represented the maximum distance. This process is then repeated for second alignment markers in each line, third alignment markers, and so on.

14 Claims, 12 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 9

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	RDNC	Dra
------	-------	----------	-------	--------	----------------	------	-----------	--------	------	-----

3. Document ID: US 5813019 A

L9: Entry 3 of 7

File: USPT

Sep 22, 1998

US-PAT-NO: 5813019

DOCUMENT-IDENTIFIER: US 5813019 A

** See image for Certificate of Correction **

TITLE: Token-based computer program editor with program comment management

DATE-ISSUED: September 22, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van De Vanter; Michael L.	Mountain View	CA		

US-CL-CURRENT: 715/512; 715/527, 715/531, 717/111

ABSTRACT:

A user-friendly text editor for structurally represented computer programs is disclosed. The present editor combines advantages of text editors and structure editors by transforming, as the user types, the text stream entered by the user into a token stream, where the tokens of the token stream constitute the words of the program being entered. Each of the tokens is classified by the editor as one of a group of extended lexemes defined by the language in which the program being edited is written. These extended lexemes are defined similar to lexemes that might be used in a batch lexer, but are more numerous as the present editor must account for the incomplete and ill-formed lexemes that arise as the user types the program. Before performing lexical analysis, the present editor separates program statements from program comments. The editor then stores the text of the comments along with comment positioning information that allows the original position of the comments as entered by the user to be recreated for printing or display purposes. Based on the token stream and the separately represented comments, the editor prettyprints the program and comments as the user types. The editor also allows the user to edit the program from the prettyprinted display as if the program were internally represented as text. The present editor also allows the user to periodically request a syntactical analysis of the program, wherein the structure of the program is represented in a syntax tree that references the token stream.

29 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) |     | [Claims](#) | [RICO](#) | [Drawn D.](#)

4. Document ID: US 5752058 A

L9: Entry 4 of 7

File: USPT

May 12, 1998

US-PAT-NO: 5752058

DOCUMENT-IDENTIFIER: US 5752058 A

** See image for Certificate of Correction **

TITLE: System and method for inter-token whitespace representation and textual editing behavior in a program editor

DATE-ISSUED: May 12, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van De Vanter, Michael L.	Mountain View	CA		

US-CL-CURRENT: 715/531; 715/514, 715/519, 715/527, 717/111

ABSTRACT:

A user-friendly editor for structurally represented computer programs is disclosed. The present editor combines advantages of text editors and structure editors by transforming, as the user types, the text stream entered by the user into a token stream, where the tokens of the token stream constitute the words of the program being entered. Each of the tokens is classified by the editor as one of group of extended lexemes defined by the language in which the program being edited is written. These extended lexemes are defined similarly to lexemes that might be used in a batch lexer, but are more numerous as the present editor must account for the incomplete and ill-formed lexemes that arise as the user types the program. Based on information in the token stream, the editor prettyprints the program as the user types. This prettyprinting step involves typesetting each of the tokens in a visually distinct manner and displaying a varying amount of visual whitespace between the tokens, each of these operations being based on the lexical types of the tokens. The editor also allows the user to edit the program from the prettyprinted display as if the program were internally represented as text via a cursor whose position and appearance depends on the lexical types of tokens adjacent to the cursor. The present editor also allows the user to periodically request a syntactical analysis of the program, wherein the structure of the program is represented in a syntax tree that references the token stream.

24 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	DOCID	Drawn
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	-------	-------

5. Document ID: US 5748975 A

L9: Entry 5 of 7

File: USPT

May 5, 1998

US-PAT-NO: 5748975

DOCUMENT-IDENTIFIER: US 5748975 A

** See image for Certificate of Correction **

TITLE: System and method for textual editing of structurally-represented computer programs with on-the-fly typographical display

DATE-ISSUED: May 5, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van De Vanter, Michael L.	Mountain View	CA		

US-CL-CURRENT: 715/531; 715/514, 715/519, 715/527, 717/111

ABSTRACT:

A user-friendly editor for structurally represented computer programs is disclosed. The present editor combines advantages of text editors and structure editors by transforming, as the user types, the event stream entered by the user into a token stream, where the tokens of the token stream constitute the words of the program being entered. Each of the tokens is classified by the editor as one of group of extended lexemes defined by the language in which the program being edited is written. These extended lexemes are defined similar to lexemes that might be used in a batch lexer, but are more numerous as the present editor must account for the incomplete and ill-formed lexemes that arise as the user types the program. Based on information in the token stream, the editor prettyprints the program as the user types. The editor also allows the user to edit the program from the prettyprinted display as if the program were internally represented as text. The present editor also allows the user to periodically request a syntactical analysis of the program, wherein the structure of the program is represented in a syntax tree that references the token stream.

33 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	Edits	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	-------	----------

6. Document ID: US 5737608 A

L9: Entry 6 of 7

File: USPT

Apr 7, 1998

US-PAT-NO: 5737608

DOCUMENT-IDENTIFIER: US 5737608 A

**** See image for Certificate of Correction ****

TITLE: Per-keystroke incremental lexing using a conventional batch lexer

DATE-ISSUED: April 7, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van De Vanter; Michael L.	Mountain View	CA		

US-CL-CURRENT: 717/112; 715/530, 715/531

ABSTRACT:

A system and method are disclosed that enable a batch lexer to be used to incrementally update a token stream representation of a computer program maintained in an editor as the computer program is being edited. A keystroke executive interprets editing inputs and dispatches editing events to a lexical analyzer. The lexical analyzer converts a range of the tokens likely to be affected to an equivalent old textual stream that preserves whitespace implied by but not represented within the token stream. A new text stream is generated from the old text stream by carrying out the current editing event. I.e., insertion of text is handled by the insertion of the relevant text into the old text stream (now the new stream) and deletion of a character is handled by deleting the appropriate

character from the old text stream. The batch lexer is then invoked on the new text stream and as a result returns a new token stream. The fewest possible tokens from the new token stream that reflect the entire impact of the current editing event are returned to the keystroke executive along with an updated editing point within the new token stream as the suggested token stream update. The keystroke executive is free to ignore or accept the suggested update. The new token stream and the new text stream can be generated lazily, respectively one token and one character at a time.

21 Claims, 11 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 11

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	10000	Drawn
------	-------	----------	-------	--------	----------------	------	-----------	-----	-----	-----	--------	-------	-------

7. Document ID: US 5586329 A

L9: Entry 7 of 7

File: USPT

Dec 17, 1996

US-PAT-NO: 5586329

DOCUMENT-IDENTIFIER: US 5586329 A

TITLE: Programmable computer with automatic translation between source and object code with version control

DATE-ISSUED: December 17, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Knudsen; Helge	Oakville			CA
Chong; Daniel T.	Woodbridge			CA
Yaffe; John	Mississauga			CA
Taugher; James E.	Mississauga			CA
Robertson; Michael	Mississauga			CA
Plazak; Zbigniew	Etobicoke			CA

US-CL-CURRENT: 717/108; 707/100, 717/112, 717/117, 717/137

ABSTRACT:

A computer which executes rules which are defined according to a language having a valid grammar. The computer comprises input means for receiving and temporarily storing a first source code representation of a rule; object code translation means for translating the first source code representation into a first object code representation executable by the computer; storage means for storing the object code representations of rules; discard means for automatically discarding from the input means the first source code representation upon the storing of the first object code representation in the storage means; source code translation means for translating the first object code representation into a second source code representation where the second source code representation has lines of text; edit means for editing the second source code representation by deleting, adding, or changing one or more of the lines of text of the second source code representation; second object code translation means for translating the second source code

representation, as edited, into a new object code representation of the edited rule for storage in the storage means; and the discard means discarding the first object code and second source code representations automatically upon the storing of the new object code representation of the edited rule in the storage means. The computer thereby minimizes the storage required in the storage means for storing rules and maintains version control over the object code representations of rules stored in the storage means.

4 Claims, 40 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 30

Display Format: REV [Change Format](#)

[Previous Page](#) [Next Page](#) [Go to Doc#](#)